

## Amendment to the Claims

1. (currently amended) A computer implemented method of for managing a database data structure that ~~includes~~ is partitioned into a plurality of sections, each of the sections comprising a plurality of data records, the method comprising:

receiving a new data record and a key that is associated with the new data record;

responsive to said receiving said new data record and said associated key, identifying one of the sections based upon the associated key of the new data record;

responsive to said identifying one of the sections, determining if said new data record fits in an unused storage space on said identified section based on a size of said new data record;

if said new data record fits in said unused storage space, then storing said new data record in said identified section;

if ~~[[a]]~~ said size of said new data record is greater than a size of said unused storage space, then ranking all data records on said identified section according to a computer implemented ranking function;

summing sizes of said all data records below rank of said new data record;

if said sum is not greater than said size of said new data record, then ending process; and

if said sum is greater than said size of said new data record, then deleting one or more data records from the identified section and storing the new data record in the identified section.

2. (canceled)

3. (currently amended) The computer implemented method of Claim 1, wherein the ranking function is a least recently used algorithm.

4. (currently amended) The computer implemented method of Claim 1, wherein the ranking function is a function of the statistical properties of the data being stored.

5. (currently amended) The computer implemented method of Claim 1, wherein  
5 each of the plurality of sections is an integer multiple of the page size that is used by an operating system to transfer data between a primary storage and a secondary storage.

6. (currently amended) The computer implemented method of Claim 1, wherein  
10 each of the sections is about the same page size that is used by an operating system to transfer data between a primary storage and a secondary storage.

7. (currently amended) The computer implemented method of Claim 1, additionally comprising allocating a contiguous memory space to contain each of the sections.

15 8. (currently amended) A program storage device storing program instructions that when executed perform the program for managing a database data structure that is partitioned into a plurality of sections, each of the sections comprising a plurality of data records, the program comprising the steps of:

receiving a new data record and a key that is associated with the new data  
20 record;

responsive to said receiving said new data record and said associated key,  
identifying one of the sections based upon the associated key of the new data record;

responsive to said identifying one of the sections, determining if said new data  
record fits in an unused storage space on said identified section based on a size of said  
25 new data record;

if said new data record fits in said unused storage space, then storing said new data record in said identified section;

if [[a]]said size of said new data record is greater than a size of said unused  
storage space, then ranking all data records on said identified section according to a  
30 computer implemented ranking function;

summing sizes of said all data records below rank of said new data record;  
if said sum is not greater than said size of said new data record, then ending  
process; and

if said sum is greater than said size of said new data record, then deleting one or  
5 more data records from the identified section and storing the new data record in the  
identified section.

9. (canceled)

10. (previously presented) The program storage device of Claim 8, wherein the  
ranking scheme identifies which ones of the data records are the least recently used.

11. (currently amended) The ~~method~~ program storage device of Claim 8, wherein  
each of the sections is about the same size that is used by an operating system to  
15 transfer data between a primary storage and a secondary storage.

12. (currently amended) A database system for managing data records, the system  
comprising:

a plurality of sections, each of the sections being about the same memory size  
20 that is used by an operating system to transfer data between a primary storage and a  
secondary storage; and

a control program which ~~receives a request for the storage of a data record, the  
control program selecting one of the sections based upon a key and storing the data  
record in the selected section;~~

25 wherein the control program:

~~determines if said data record fits in an unused space on said selected  
section;~~

~~if said data record fits in said unused space, then stores said data record in  
said selected section;~~

~~if a size of said data record is greater than a size of said unused space, then~~  
~~rank all data records on said selected section according to a ranking function;~~

~~sums sizes of said all data records below rank of said data record;~~

~~if said sum is not greater than said size of said data record, then ends~~  
5 ~~process; and~~

~~if said sum is greater than said size of said data record, then deletes one or~~  
~~more data records from the selected section and stores the data record in the~~  
~~selected section. performing the steps of:~~

~~receiving a new data record and a key that is associated with the new data~~  
10 ~~record;~~

~~responsive to said receiving said new data record and said associated key,~~  
~~identifying one of the sections based upon the associated key of the new data record;~~

~~responsive to said identifying one of the sections, determining if said new data~~  
~~record fits in an unused storage space on said identified section based on a size of said~~  
15 ~~new data record;~~

~~if said new data record fits in said unused storage space, then storing said new~~  
~~data record in said identified section;~~

~~if said size of said new data record is greater than a size of said unused storage~~  
~~space, then ranking all data records on said identified section according to a computer~~  
20 ~~implemented ranking function;~~

~~summing sizes of said all data records below rank of said new data record;~~

~~if said sum is not greater than said size of said new data record, then ending~~  
~~process; and~~

~~if said sum is greater than said size of said new data record, then deleting one or~~  
25 ~~more data records from the identified section and storing the new data record in the~~  
~~identified section.~~

13. (canceled)

14. (previously presented) The database system of Claim 12, wherein the ranking function determines a last access time for each of the data records or the selected sections.

5 15. (original) The database system of Claim 12, wherein at least one of the sections includes at least one item of section information.

16. (original) The database system of Claim 15, wherein the section information includes the number of data records that are contained in the section.

10

17. (original) The database system of Claim 15, wherein the section information includes an offset from the beginning of the section to the first unused position within the section.

15 18. (original) The database system of Claim 15, wherein the section information includes a section number that is associated with the section.

19. (original) The database system of Claim 12, additionally comprising a client application which provides the storage request of the data record and the key to the control program.

20

20. (currently amended) A database system for managing information items, the system comprising:

a plurality of sections; and

25 a control program performing the steps of:

~~which receives~~ receiving a request for the storage of a data record[.], and then  
~~the control program selecting one of the plurality of sections; and storing the data record in the selected section, the control program:~~

responsive to said selecting one of the plurality of sections, determining if said

30 data record fits in an unused space on said selected section;

responsive to if said data record fits in said unused space, then storing said data record in said selected section~~[[:]]~~, otherwise

~~determining if a size of said data record is greater than a size of said unused space, then ranking all data records on said selected section removing one or more~~

5 selected data records according to a ranking function~~[[:]]~~.

~~summing sizes of said all data records below rank of said data record;~~

~~if said sum is not greater than said size of said data record, then ending process;~~

and

10 ~~if said sum is greater than said size of said data record, then deleting one or more data records from the selected section and storing the data record in the selected section.~~

21. (original) The database system of Claim 20, wherein the ranking function determines an access time for each of the data records or the selected sections.

15

22. (original) The database system of Claim 20, wherein each of the data records stores at least one user profile.

20 23. (original) The database system of Claim 20, wherein at least one of the sections includes at least one record of section information.

24. (original) The database system of Claim 23, wherein the section information includes the number of data records that are contained in the section.

25 25. (original) The database system of Claim 23, wherein the section information includes an offset from the beginning of the section to the first unused position within the section.

30 26. (original) The database system of Claim 23, wherein the section information includes a section number that is associated with the section.

27. (original) The database system of Claim 20, additionally comprising a client application which provides the storage request of the data record and the key to the control program.

5

28. (original) The database system of Claim 20, wherein the size of each of the sections is an integer multiple to the transfer size that is used by an operating system to transfer data between a primary storage and a secondary storage.

10

29. (original) The database system of Claim 20, wherein the size of each of the sections is about equal to the transfer size that is used by an operating system to transfer data between a primary storage and a secondary storage.

15

30. (currently amended) A system for managing a database that ~~includes is~~ partitioned into a plurality of sections, each of the sections comprising a plurality of data records, the system comprising:

means for receiving one or more new data records, each of the new data records having an associated key;

responsive to said receiving said new data record and said associated key,

20

means for identifying one of the sections based upon the associated key of the new data record;

responsive to said identifying one of the sections, means for determining if said new data record fits in an unused space on said identified section based on a size of said new data record;

25

if said new data record fits in said unused storage space, then means for storing said new data record in said identified section;

if [[a]]said size of said new data record is greater than a size of said unused storage space, then means for ranking all data records on said identified section according to a computer implemented ranking function;

means for summing sizes of said all data records below rank of said new data record;

if said sum is not greater than said size of said new data record, then means for ending process; and

5 if said sum is greater than said size of said new data record, then means for deleting one or more data records from said identified section and means for storing said new data record in the identified section.

31. (canceled)

10

32. (original) The system of Claim 30, wherein the ranking function identifies which ones of the data records that are the least recently used.

33. (original) The system of Claim 30, wherein the database occupies a single  
15 contiguous physical memory space.

34. (original) The system of Claim 30, wherein the size of each of the sections is an integer multiple to the page size that is used by an operating system to transfer data between a primary storage and a secondary storage.

20

35. (original) The system of Claim 30, wherein the size of each of the sections is about equal to the page size that is used by an operating system to transfer data between a primary storage and a secondary storage.

25 36. (currently amended) A database system for managing information records, the system comprising:

a primary storage;

a secondary storage having a plurality of pages;



a plurality of sections, wherein each of the sections is adapted to contain one or more data records, and wherein each of the sections resides in the secondary storage on one of the plurality of pages; and

a control program which receives a request for the retrieval of a data record, the  
5 control program retrieving the data record from the secondary storage and for storing the data record in the primary storage, wherein the retrieval operation reads at most one page from the secondary storage, and wherein the control program further performs the steps of:[:]

~~— a database data structure having a plurality of sections, each of the sections  
10 residing on one of the pages in any of: the primary storage and the secondary storage; and~~

~~a database manager which receives requests from the client application to store a data record in the database data structure, wherein the database manager selects one of the sections and stores the data record in the selected section, wherein the  
15 database manager:~~

~~determines if said data record fits in an unused space on said selected section;~~

~~if said data record fits in said unused space, then stores said data record in said selected section;~~

20 ~~determines if a size of said data record is greater than a size of said unused space, then ranks all data records on said selected section according to a ranking function;~~

~~sums sizes of said all data records below rank of said data record;~~

~~if said sum is not greater than said size of said data record, then ends  
25 process; and~~

~~if said sum is greater than said size of said data record, then deletes one or more data records from the selected section and stores the data record in the selected section.~~

receiving a new data record and a key that is associated with the new data  
30 record;

responsive to said receiving said new data record and said associated key,  
identifying one of the sections based upon the associated key of the new data record;

responsive to said identifying one of the sections, determining if said new data  
record fits in an unused storage space on said identified section based on a size of said  
5 new data record;

if said new data record fits in said unused storage space, then storing said new  
data record in said identified section;

if said size of said new data record is greater than a size of said unused storage  
space, then ranking all data records on said identified section according to a computer  
10 implemented ranking function;

summing sizes of said all data records below rank of said new data record;

if said sum is not greater than said size of said new data record, then ending  
process; and

if said sum is greater than said size of said new data record, then deleting one or  
15 more data records from the identified section and storing the new data record in the  
identified section.

37. (currently amended) A database system for managing information items, the  
system comprising:

20 a primary storage;

a secondary storage having a plurality of pages;

a plurality of sections, each of the sections independent of each of the other  
sections such that an error in one of the sections does not affect any of the other  
sections; and

25 a control program which receives a request for the retrieval of a data record, the  
control program retrieving the data record from the secondary storage so that the data  
record is stored in the primary storage and wherein said control program performs the  
steps of:[;]]

~~a database data structure having a plurality of sections, each of the sections residing on one of the pages in any of: the primary storage and the secondary storage; and~~

~~a database manager which receives requests from the client application to store  
5 a data record in the database data structure, wherein the database manager selects one of the sections and stores the data record in the selected section, wherein the database manager:~~

~~determines if said data record fits in an unused space on said selected section;~~

~~10 if said data record fits in said unused space, then stores said data record in said selected section;~~

~~determines if a size of said data record is greater than a size of said unused space, then ranks all data records on said selected section according to a ranking function;~~

~~15 sums sizes of said all data records below rank of said data record;~~

~~if said sum is not greater than said size of said data record, then ends process; and~~

~~if said sum is greater than said size of said data record, then deletes one or more data records from the selected section and stores the data record in the  
20 selected section.~~

receiving a new data record and a key that is associated with the new data record;

responsive to said receiving said new data record and said associated key, identifying one of the sections based upon the associated key of the new data record;

~~25 responsive to said identifying one of the sections, determining if said new data record fits in an unused storage space on said identified section based on a size of said new data record;~~

if said new data record fits in said unused storage space, then storing said new data record in said identified section;

if said size of said new data record is greater than a size of said unused storage space, then ranking all data records on said identified section according to a computer implemented ranking function;

summing sizes of said all data records below rank of said new data record;

5 if said sum is not greater than said size of said new data record, then ending process; and

if said sum is greater than said size of said new data record, then deleting one or more data records from the identified section and storing the new data record in the identified section.

10

38. (currently amended) A database system for managing information items, the system comprising:

a client application;

a primary storage comprising a plurality of pages;

15 a secondary storage comprising a plurality of pages;

a caching subsystem for copying pages from the secondary storage to the pages in the primary storage and vice-versa;

a database data structure having a plurality of sections, each of the sections residing on one of the pages in any of: the primary storage and the secondary storage;

20 and

a database manager which receives requests from the client application to store a data record in the database data structure, wherein the database manager performs the steps of: selects one of the sections and stores the data record in the selected section, wherein the database manager:

25 determines if said data record fits in an unused space on said selected section;

if said data record fits in said unused space, then stores said data record in said selected section;

~~determines if a size of said data record is greater than a size of said unused space, then ranks all data records on said selected section according to a ranking function;~~

~~sums sizes of said all data records below rank of said data record;~~

5 ~~if said sum is not greater than said size of said data record, then ends process; and~~

~~if said sum is greater than said size of said data record, then deletes one or more data records from the selected section and stores the data record in the selected section.~~

10 receiving a new data record and a key that is associated with the new data record;

responsive to said receiving said new data record and said associated key, identifying one of the sections based upon the associated key of the new data record;

15 responsive to said identifying one of the sections, determining if said new data record fits in an unused storage space on said identified section based on a size of said new data record;

if said new data record fits in said unused storage space, then storing said new data record in said identified section;

20 if said size of said new data record is greater than a size of said unused storage space, then ranking all data records on said identified section according to a computer implemented ranking function;

summing sizes of said all data records below rank of said new data record;

if said sum is not greater than said size of said new data record, then ending process; and

25 if said sum is greater than said size of said new data record, then deleting one or more data records from the identified section and storing the new data record in the identified section.